REVAMP: AUTOMATED SIMULATIONS OF ADVERSARIAL ATTACKS ON ARBITRARY OBJECTS IN REALISTIC SCENES

Matthew Hull, Zijie. J. Wang & Duen Horng Chau
Georgia Institute of Technology Atlanta, GA 30332, USA
{matthewhull,jayw,polo}@gatech.edu

ABSTRACT

Deep learning models, such as those used in autonomous vehicles, are vulnerable to adversarial attacks where attackers could place adversarial objects in the environment to induce incorrect detections. While generating such adversarial objects in the digital realm are well-studied, successfully transferring these attacks to the physical realm remains challenging, especially when accounting for real-world environmental factors. We address these challenges with REVAMP, a first-of-its-kind Python library for creating attack scenarios with arbitrary objects in scenes with realistic environmental factors, lighting, reflection, and refraction. REVAMP empowers researchers and practitioners to swiftly explore diverse scenarios, offering a wide range of configurable options for experiment design and using differentiable rendering to replicate physically-plausible adversarial objects. REVAMP is open-source and available at https://github.com/poloclub/revamp and a demo video is available at https://youtu.be/NA0XR0XkS1E.

1 INTRODUCTION

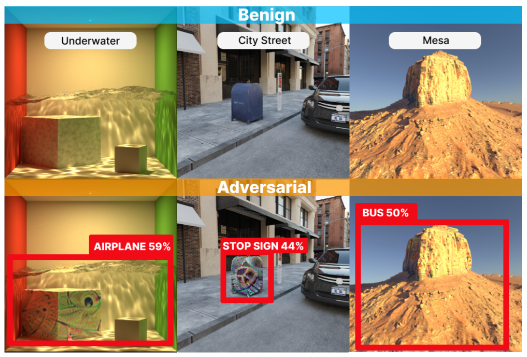
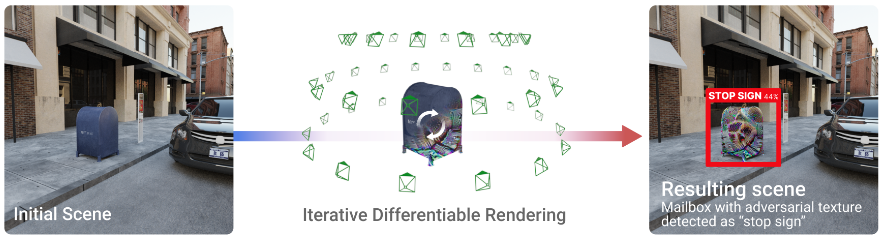


Fig. 1: REVAMP enables users to easily perturb textures on objects in realistic scenes (top row), inducing incorrect detection of an underwater cube as an airplane, a mailbox on a city street as a stop sign, and a mesa as a bus (bottom row).

Deep neural networks used for image recognition and object detection are vulnerable to adversarial attacks Goodfellow et al. (2015). While initially confined to the digital realm, there is a growing interest in exploring physically realizable adversarial attacks Chen et al. (2018). Recent efforts have investigated methods like printing adversarial textures onto objects Chen et al. (2018), projection mapping Huang & Ling (2022), and marker and sticker placement Song et al. (2018) to create physical adversarial objects. However, successfully transferring attacks from the digital to the physical realm remains challenging. Techniques like Expectation over Transformation (EoT) Athalye et al. (2018) focus mainly on limited scaling and rotation transformations, often overlooking real-world factors such as environmental conditions and lighting. Experimentation in 3D environments that realistically portray the real world has strong potential to address the these limitations Xiao et al. (2019). However, creating such environments require expertise in 3D modeling and rendering. Furthermore, traditional 3D rendering lacks support for optimizing scene parameters and textures due to non-differentiable operations. We address these challenges by making two major contributions:

- **We introduce REVAMP, the first-of-its-kind open-source Python library for easily creating attack scenarios with arbitrary objects in realistic scenes.** REVAMP empowers researchers and practitioners to swiftly explore diverse attack scenarios, providing a customizable adversarial attack pipeline with configurable options for designing experiments and creating 3D objects. Leveraging *differentiable rendering*, REVAMP optimizes scene parameters by propagating gradients through

With a single REVAMP command, the mailbox in a realistic scene is iteratively perturbed via differentiable rendering to induce a stop sign misdetection across camera positions


Initial Scene
Iterative Differentiable Rendering
STOP SIGN 44%
Resulting scene
Mailbox with adversarial texture detected as "stop sign"

```
python revamp.py scene=city texture=mail_box attack_class=stop_sign multicam=64
```

Fig. 2: REVAMP's pipeline can be run with a simple command, allowing a user to choose a scene, an object to attack, the desired attack class, and the number of camera positions to use. Differentiable rendering is used to iteratively render and update the texture of the attacked object.

the rendering process to induce attacks Jakob et al. (2022). Users can select from scenes that replicate real-world scenarios, or create their own, specifying textures and parameters for crafting tailored adversarial examples. Our library of scenes supports realistic lighting effects like reflection and refraction, and other realistic environmental factors (Fig. 1). REVAMP stands for ***R**eshaping **E**nvironment by **V**arying **A**dversarial **M**odel's **P**arameters*.

- **Open-Source Implementation:** While some prior works Liu et al. (2019); Xiao et al. (2019) have explored differentiable rendering for adversarial examples, they are closed-source. REVAMP is open-source, `https://github.com/poloclub/revamp`, facilitating reproducible research for machine learning (ML) security researchers and practitioners.

2 System Design and Implementation

REVAMP uses the Hydra configuration management tool Yadan (2019) to construct and customize scenarios, providing flexibility by allowing parameter adjustments through the command line. Each scenario is described in a YAML configuration file, which specifies a 3D scene, an attackable parameter, rendering settings, and a victim model (Faster R-CNN Ren et al. (2015)). For example, Figure 2 shows a "city" scene with a mailbox's texture as the "attackable" parameter. The scenario configuration specifies the inference model, optimizable parameters, and attack type (targeted or untargeted). Camera transformations, such as orbiting, can also be configured, offering diverse perspectives for exploring adversarial impacts. REVAMP employs the Mitsuba differentiable renderer with ray-tracing to precisely simulate real-world phenomena, enabling flexible evaluation of adversarial attack scenarios that can be easily modified (e.g., add/delete meshes, textures, light sources) that have only recently begun to be explored by other techniques, such as NeRF. We do currently not restrict the type of model or attack type, and REVAMP could be adapted in the future to support other tasks, such as semantic segmentation.

3 Usage Scenario

Jim is an ML security engineer at a self-driving car company. He wants to leverage REVAMP to assess and enhance the object detector's robustness for potential deployment in self-driving cars. Interested in investigating how large objects in a street environment might be misclassified, if manipulated adversarially, Jim selects a city scene and designates the mailbox's texture as the attackable parameter.

To conduct this simulation, Jim executes REVAMP's pipeline with a simple Python command to specify the city scene, the mailbox texture and the desired misdetection target class, e.g., "stop sign" (Fig. 2). At each step of differentiable rendering, the resulting scene image is forwarded to the object detector, and the texture is iteratively perturbed using a PGD ℓ_2 attack Madry et al. (2018) until a "stop sign" detection is achieved. REVAMP outputs rendered images, object detection bounding boxes, class labels, and perturbed texture maps for each rendering step. Jim notes substantial perturbation was required before a "stop sign" detection with low confidence was achieved.

Following the pipeline run, Jim has gained confidence that the object detector is resistant against a strong perturbation, and proceeds to use the output texture maps to conduct adversarial training to further strengthen model robustness. The ease of use of REVAMP encourages Jim to quickly expand his experiment to more realistic scenes such as open highways or unmarked roads in wooded areas.

URM Statement

The authors acknowledge that at least one key author of this work meets the URM criteria of ICLR 2024 Tiny Papers Track.

References

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 284–293. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/athalye18b.html.

Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Polo Chau. Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 52–68. Springer, 2018.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6572.

Bingyao Huang and Haibin Ling. Spaa: Stealthy projector-based adversarial attacks on deep image classifiers. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 534–542, 2022. doi: 10.1109/VR51125.2022.00073.

Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, Merlin Nimier-David, Delio Vicini, Tizian Zeltner, Baptiste Nicolet, Miguel Crespo, Vincent Leroy, and Ziyi Zhang. Mitsuba 3 renderer, 2022. https://mitsuba-renderer.org.

Hsueh-Ti Derek Liu, Michael Tao, Chun-Liang Li, Derek Nowrouzezahrai, and Alec Jacobson. Beyond pixel norm-balls: Parametric adversaries using an analytically differentiable renderer. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SJl2niR9KQ.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pp. 91–99, Cambridge, MA, USA, 2015. MIT Press.

Dawn Song, Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramèr, Atul Prakash, and Tadayoshi Kohno. Physical adversarial examples for object detectors. In *12th USENIX Workshop on Offensive Technologies (WOOT 18)*, Baltimore, MD, August 2018. USENIX Association. URL https://www.usenix.org/conference/woot18/presentation/eykholt.

Chaowei Xiao, Dawei Yang, Bo Li, Jia Deng, and Mingyan Liu. Meshadv: Adversarial meshes for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Omry Yadan. Hydra - a framework for elegantly configuring complex applications. Github, 2019. URL https://github.com/facebookresearch/hydra.